UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/994,574	11/26/2001	William R. Wheeler	10559/602001/P12886	7301

20985 7590 12/06/2007
FISH & RICHARDSON, PC
P.O. BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
GUILL, RUSSELL L	

ART UNIT	PAPER NUMBER
2123	

MAIL DATE	DELIVERY MODE
12/06/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/994,574	WHEELER ET AL.	
	Examiner	Art Unit	
	Russ Guill	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 November 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,4-7,10-13,15,18,21-23 and 25-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4-7,10-13,15,18,21-23 and 25-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 November 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to an Amendment filed November 13, 2007. No claims were added or cancelled. Claims 1, 4 – 7, 10 – 13, 15, 18, 21 – 23 and 25 - 27 are pending. Claims 1, 4 – 7, 10 – 13, 15, 18, 21 – 23 and 25 - 27 have been examined. Claims 1, 4 – 7, 10 – 13, 15, 18, 21 – 23 and 25 - 27 have been rejected.

2. **As previously recited, the Examiner would like to thank the Applicant for the well-presented response, which was useful in the examination process.** The Examiner appreciates the effort to perform a careful analysis of the Office Action and make very well presented arguments and amendments.

3. As an initial matter, the Examiner would like to respectfully direct the Applicant to the art recited in the Conclusion section that teaches knowledge of the ordinary artisan, and appears to be relevant to the Applicant's specification.

Response to Arguments

4. Regarding **claims 1 and 15** rejected under 35 U.S.C. § 101:

   4.1. Applicants' arguments have been fully considered, and are persuasive.

5. Regarding **claim 1** objected to for informalities:

   5.1. Applicants' arguments have been fully considered, and are persuasive.

6. Regarding **claim 15** rejected under 35 U.S.C. § 112, second paragraph:

   6.1. Applicants' arguments have been fully considered, and are persuasive.

7. Regarding **claims** rejected under 35 U.S.C. § 103:

7.1. Applicants' arguments have been fully considered, and are persuasive. However, the Examiner respectfully disagrees with parts of Applicant's arguments as described below. Further, after further consideration and search, new rejections are made below, necessitated by the amended limitations.

7.2. The Applicant argues:

7.3. As amended, claim 1 relates to a system that includes a logic design module operable on one or more data processing devices to be used by one or more users to generate a logic design as part of an electrical circuit and a central database integrated with the logic design module and including a collection of modifiable values of signal parameters that are accessible by the logic design module. The logic design includes labels. The values of signal parameters are associated with the labels in the logic design. The logic design module is operable to update the logic design to reflect modification of the values of the signal parameters in the central database by modifying the logic design to be compatible with the modified values of the signal parameters, and to indicate design discrepancies in the logic design resulting from the modifications to the values of the signal parameters in the central database automatically. The collection of modifiable values is separate from the logic design.

7.4. The rejection contends that it would have been obvious for one of ordinary skill to combine Panchul, Yamagishi, and the IEEE Std. 1364-1995 to arrive at the subject matter recited in claim 1.

7.5. Applicant respectfully disagrees. In this regard, Panchul describes a system for converting a high-level language program to a hardware design implementation. *See, e.g., Panchul,* col. 4, line 3-8. A user initially renders a preliminary hardware design in a standard high-level programming language. *See, e.g., Panchul,* col. 12, line 34-37. *See also Panchul,* col. 12, line 51-67. Panchul's system then compiles the high-level programming language into a hardware description language (HDL) synthesizable design. *See,. e.g., Panchul,* col. 13, line 33-37. The HDL design is then synthesized into a gate-level

hardware representation. *See, e.g., Panchul,* col. 13, line 53-57. *See also Panchul,* col. 12, line 42-45. In turn, the gate-level hardware representation can be reduced to an actual hardware implementation. *See, e.g., Panchul,* col. 13, line 64-col. 14, line 16; col. 12, line 45-47.

7.6. Panchul neither describes nor suggests that a collection of modifiable values separate from the logic design are associated with labels in either Panchul's preliminary hardware designs or HDL designs. In this regard, as discussed above, Panchul's preliminary hardware designs are in a standard high-level programming language. *See also Panchul,* col. 14, line 16-35. There is no description or suggestion in Panchul that a collection of modifiable values that are associated with labels in Panchul's preliminary hardware designs is separate from the preliminary hardware designs. Instead, Panchul's preliminary hardware designs are understood to be self-contained in standard high-level programming language.

7.6.1. The Examiner respectfully replies:

7.6.2. The Applicant asserts that Panchul neither describes nor suggests that a collection of modifiable values separate from the logic design are associated with labels in either Panchul's preliminary hardware designs or HDL designs. While Panchul may not describe that a collection of modifiable values separate from the logic design are associated with labels in Panchul's preliminary hardware designs, Panchul does appear to describe a collection of modifiable values separate from the logic design are associated with labels in Panchul's HDL designs. As shown in figure 18B1, the elements ADDRESS_SIZE and DATA_SIZE are clearly labels as used in the Verilog HDL design. Further, under a broad interpretation, the labels ADDRESS_SIZE and DATA_SIZE are separate from the logic design since they are part of the "Global macro definitions", and not part of a module that contains logic design. However, better references are used in the rejection below to teach the amended limitation.

7.7. The Applicant argues:

7.8. Moreover, there is no description or suggestion in Panchul that such labels are somehow inserted into a logic design when the preliminary

hardware designs are compiled into an HDL design or that a collection of modifiable values is somehow created. Indeed, FIGS. 3-27 of Panchul show numerous examples of the compilation of high-level C-type programs into HDL representations. *See, e.g., Panchul,* col. 8, line 5-col. 11, line 37. None of these are understood to describe or suggest that labels are inserted into the hardware designs represented preliminarily in the standard high-level programming languages, or that a separate collection of modifiable values of signal parameters that are associated with such labels is somehow created.

7.8.1. The Examiner respectfully replies:

7.8.2. The Applicant asserts that there is no description or suggestion in Panchul that such labels are somehow inserted into a logic design when the preliminary hardware designs are compiled into an HDL design or that a collection of modifiable values is somehow created. The Examiner respectfully disagrees: As shown in figure 18B1, the elements ADDRESS_SIZE and DATA_SIZE are clearly labels as used in the Verilog HDL design, and they represent a collection of modifiable values.

7.9. The Applicant argues:

7.10. The rejection is understood to point to the definitions of ADDRESS_SIZE and DATA SIZE in Panchul's FIG. 18B1 as allegedly showing such labels. Applicant respectfully disagrees. In standard high-level programming languages, such as illustrated in Panchul's FIG. 18B1, such definitions are part of the code. In Panchul, this code is the preliminary hardware design. The ADDRESS_SIZE and DATA_SIZE definitions thus do not label the preliminary hardware design and are not separate from the preliminary hardware design.

7.10.1. The Examiner respectfully replies:

7.10.2. The Examiner respectfully disagrees with the Applicant's assertion that the definitions of ADDRESS_SIZE and DATA SIZE in Panchul's FIG. 18B1 do not show such labels. The Applicant appears to argue that the ADDRESS_SIZE and DATA_SIZE definitions do not label the preliminary hardware design and are not

separate from the preliminary hardware design, and then concludes that the definitions of ADDRESS_SIZE and DATA SIZE in Panchul's FIG. 18B1 do not show such labels. The conclusion does not appear to follow from the premise, since as shown in figure 18B1, the elements ADDRESS_SIZE and DATA_SIZE are clearly labels as used in the Verilog HDL design, and they represent a collection of modifiable values.

7.11. The Applicant argues:

7.12. Further, since Panchul's system compiles such a unlabled high-level programming language into an HDL design, Applicant understands the HDL designs formed by such compiling to also be unlabeled. Accordingly, Panchul's HDL designs also do not describe or suggest that a logic design include labels or that a collection of modifiable values separate from Panchul's HDL designs are associated with such labels.

7.12.1. The Examiner respectfully replies:

7.12.2. The Examiner respectfully disagrees with the Applicant's conclusion that Panchul's HDL designs do not describe or suggest that a logic design include labels or that a collection of modifiable values separate from Panchul's HDL designs are associated with such labels. The Applicant appears to argue that because the high-level programming language is unlabeled, that the HDL designs are also unlabeled, but this conclusion does not appear to follow from the premise. The conclusion does not appear to follow from the premise, since as shown in figure 18B1, the elements ADDRESS_SIZE and DATA_SIZE are clearly labels as used in the Verilog HDL design, and they represent a collection of modifiable values. Further, under a broad interpretation, the labels ADDRESS_SIZE and DATA_SIZE are separate from the logic design since they are part of the "Global macro definitions", and not part of a module that contains logic design. However, better references are used in the rejection below to teach the amended limitation.

7.13. The Applicant argues:

7.14. Finally, Panchul's synthesis of the HDL design into a gate-level hardware representation and reduction of the gate-level hardware representation into an actual hardware implementation are "conventional" and "do not constitute [Panchul's] invention per se." *See, e.g., Panchul,* col. 12, line 45-47; col. 13, line 53-57; col. 13, line 60-61. Applicant also submits that nothing in Panchul's synthesis and reduction of gate-level hardware representations describes or suggests that a logic design include labels or that a separate collection of modifiable values are associated with such labels.

7.14.1. The Examiner respectfully replies:

7.14.2. The rejection does not appear to rely upon Panchul's synthesis of the HDL design into a gate-level hardware representation and reduction of the gate-level hardware representation into an actual hardware implementation to teach the limitation.

7.15. The Applicant argues:

7.16. Yamagishi and the IEEE Std. 1364-1995 do not remedy these deficiencies in Panchul. Indeed, Yamagishi at least suggests that an exhaustive array of different design data be maintained in a single "design entity." *See, e.g., Yamagishi,* § 2.1. Accordingly, nothing in Yamagishi would lead one of ordinary skill to modify Panchul so that a collection of modifiable values separate from a logic design are associated with labels in a logic design, as recited in claim 1.

7.16.1. The Examiner respectfully replies:

7.16.2. The rejection does not appear to rely upon Yamagishi to teach the recited limitation.

7.17. The Applicant argues:

7.18. As for the cited portion of the IEEE Std. 1364-1995, this portion deals with specifying a range of repetitive instances. These specifications are understood to be part of the code that constitutes a hardware design. Accordingly, nothing in IEEE Std. 1364-1995 would lead one of ordinary skill to modify Panchul so that a collection of modifiable values separate from a logic design are associated with labels in the logic design, as recited in claim 1.

7.18.1. The Examiner respectfully replies:

7.18.2. The previous Office Action does not appear to rely upon IEEE to teach the recited limitation.

7.19. The Applicant argues:

7.20. Thus, even if Panchul, Yamagishi, and the IEEE Std. 13641995 were combined, one of ordinary skill would not arrive at a collection of modifiable values of signal parameters that is separate from the logic design and that are accessible by a logic design module to update the logic design to reflect modification thereof. Accordingly, claim 1 is not obvious over Panchul, Yamagishi, and the IEEE Std. 1364-1995. Applicant respectfully requests that the rejections of claim 1 and the claims dependent therefrom be withdrawn.

7.20.1. The Examiner respectfully replies:

7.20.2. A new rejection is made below to teach the amended limitation.

7.21. The Applicant argues:

7.22. Claim 7 was rejected under 35 U.S.C. § 103(a) as obvious over Panchul, Yamagishi, and IEEE Std. 1364-1995.

7.23. As amended, claim 7 relates to a computer-implemented method that includes receiving an assignment of a value to a signal parameter, maintaining the value of the signal parameter in a central database in association with an identifier of the signal parameter, using the identifier of the signal parameter

maintained in the central database to identify a first position in computer code for a logic design forming part of an electrical circuit, modifying the computer code at the first position to reflect the value, using the identifier of the signal parameter maintained in the central database to identify a second position in the computer code for the logic design, modifying the computer code at the second position to reflect the value, receiving an updated value of the signal parameter in the central database, updating both the first position and the second position in the computer code for the logic design to reflect the updated value of the signal parameter, and indicating design discrepancies occurring in the logic design that result from updating the value of the defined signal parameter automatically. The value of the signal parameter is maintained separately from the computer code for the logic design.

7.24. Panchul, Yamagishi, and IEEE Std. 1364-1995 neither describe nor suggest modifying computer code for a logic design at first and second positions to reflect the value of a signal parameter maintained separately from the computer code for the logic design in a central database, as recited in claim 7.

7.25. In this regard, as discussed above, Panchul's preliminary hardware designs are understood to be self-contained in standard high-level programming language. Panchul's HDL designs and gate-level hardware representations are produced from the self-contained preliminary hardware designs and also believed to be self-contained.

7.26. Thus, Panchul neither describes nor suggests modifying computer code for a logic design at first and second positions to reflect the value of a signal parameter maintained separately from the computer code for the logic design in a central database, as recited in claim 7.

7.26.1. The Examiner respectfully replies:

7.26.2. The Examiner respectfully disagrees with the Applicant's conclusion that Panchul neither describes nor suggests modifying computer code for a logic design at first and second positions to reflect the value of a signal parameter maintained separately from the computer code for the logic design in a central database. This conclusion does not appear to follow from the Applicant's premise. Since figure

18B1 shows Verilog HDL code, it would have been obvious that when the code was compiled, that the instances of DATA_SIZE in the code would be updated with the value of the label DATA_SIZE, which appears to be consistent with the operation of the invention in the Applicant's specification. Further, under a broad interpretation, the labels ADDRESS_SIZE and DATA_SIZE are separate from the logic design since they are part of the "Global macro definitions", and not part of a module that contains logic design. However, better references are used in the rejection below to teach the amended limitation.

7.27. The Applicant argues:

7.28. Yamagishi and the IEEE Std. 1364-1995 do not remedy these deficiencies in Panchul. Indeed, Yamagishi at least suggests that an exhaustive array of different design data be maintained in a single "design entity," while the cited portion of IEEE Std. 1364-1995 deals with specifications of ranges of repetitive instances in the code that constitutes a hardware design. Accordingly, nothing in Yamagishi and the IEEE Std. 1364-1995 would lead one of ordinary skill to modify computer code for a logic design at first and second positions to reflect the value of a signal parameter maintained separately from the computer code for the logic design in a central database, as recited in claim 7.

7.28.1. The Examiner respectfully replies:

7.28.2. The previous Office Action does not appear to rely upon Yamagishi and IEEE to teach the recited limitation.

7.29. The Applicant argues:

7.30. Thus, even if Panchul, Yamagishi, and the IEEE Std. 13641995 were combined, one of ordinary skill would not arrive at the recited subject matter. Accordingly, claim 7 is not obvious over Panchul, Yamagishi, and the IEEE Std. 1364-1995. Applicant respectfully requests that the rejections of claim 7 and the claims dependent therefrom be withdrawn.

7.30.1. The Examiner respectfully replies:

7.30.2. A new rejection is made below to teach the amended limitation.

7.31. The Applicant argues:

7.32. Claim 15 was rejected under 35 U.S.C. § 103(a) as obvious over Panchul and Yamagishi, and IEEE Std. 1364-1995.

7.33. As amended, claim 15 relates to an apparatus that includes a central database accessible by one or more users, modification logic operable on one or more data processing devices to allow a user to modify the values associated with identifiers individually, and an interface to convey the identifiers and the associated values from the central database to a logic design module that is operable on one or more data processing devices and that uses the identifiers to identify where a logic design is to be changed and the values to change a bit width in the logic design to form part of an electrical circuit. The central database includes a collection of identifiers of one or more bit width signal parameters and values associated with each of the identifiers of the bit width signal parameters. The collection of identifiers of one or more bit width signal parameters is maintained separately from the logic design.

7.34. Panchul and Yamagishi neither describe nor suggest a central database that includes a collection of identifiers of one or more bit width signal parameters and values associated with each of the identifiers, wherein the collection of identifiers is maintained separately from a logic design, as recited in claim 15.

7.34.1. The Examiner respectfully replies:

7.34.2. The Examiner respectfully disagrees with the Applicant's assertion that Panchul and Yamagishi neither describe nor suggest a central database that includes a collection of identifiers of one or more bit width signal parameters and values associated with each of the identifiers, wherein the collection of identifiers is maintained separately from a logic design. As described in the rejection below, Panchul and Yamagishi appear to teach a central database that includes a collection of identifiers of one or more bit width signal parameters and values associated with each of the identifiers. Further, under a broad interpretation, in

Panchul figure 18B1, the labels ADDRESS_SIZE and DATA_SIZE are separate from the logic design since they are part of the "Global macro definitions", and not part of a module that contains logic design. Therefore, Panchul appears to teach the limitation that the collection of identifiers is maintained separately from a logic design. However, better references are used in the rejection below to teach the amended limitation.

7.35. The Applicant argues:

7.36. In this regard, as discussed above, Panchul's preliminary hardware designs are understood to be self-contained in standard high-level programming language. Panchul's HDL designs and gate-level hardware representations are produced from the self-contained preliminary hardware designs and also believed to be self-contained.

7.37. Thus, Panchul neither describes nor suggests a collection of identifiers of one or more bit width signal parameters and values associated with each of the identifiers, wherein the collection of identifiers is maintained separately from a logic design, as recited in claim 15.

7.37.1. The Examiner respectfully replies:

7.37.2. The Examiner respectfully disagrees with the Applicant's conclusion that Panchul neither describes nor suggests a collection of identifiers of one or more bit width signal parameters and values associated with each of the identifiers, wherein the collection of identifiers is maintained separately from a logic design. The conclusion does not appear to follow from the premise. Further, as shown in figure 18B1, the elements ADDRESS_SIZE and DATA_SIZE are clearly identifiers of one or more bit width signal parameters and have values associated with each of the identifiers. Further, under a broad interpretation, the labels ADDRESS_SIZE and DATA_SIZE are maintained separate from the logic design since they are part of the "Global macro definitions", and not part of a module that contains logic design. However, better references are used in the rejection below to teach the amended limitation.

7.38. The Applicant argues:

7.39. Yamagishi does not remedy these deficiencies in Panchul. Indeed, Yamagishi at least suggests that an exhaustive array of different design data be maintained in a single "design entity." Accordingly, nothing in Yamagishi would lead one of ordinary skill to a collection of identifiers of one or more bit width signal parameters and values associated with each of the identifiers, wherein the collection of identifiers is maintained separately from a logic design, as recited in claim 15.

7.39.1. The Examiner respectfully replies:

7.39.2. The rejection does not appear to rely upon Yamagishi to teach the recited limitation.

7.40. The Applicant argues:

7.41. Thus, even if Panchul and Yamagishi were combined, one of ordinary skill would not arrive at the recited subject matter. Accordingly, claim 15 is not obvious over Panchul and Yamagishi. Applicant respectfully requests that the rejections of claim 15 and the claims dependent therefrom be withdrawn.

7.41.1. The Examiner respectfully replies:

7.41.2. A new rejection is made below to teach the amended limitation.

7.42. The Applicant argues:

7.43. Claim 18 was rejected under 35 U.S.C. § 103(a) as obvious over Panchul, Yamagishi, and IEEE Std. 1364-1995.

7.44. As amended, claim 18 relates to a machine-accessible medium containing instructions which cause a machine to perform operations. The operations include receiving a value of a signal parameter that characterizes multiple bits of a multiple bit signal, maintaining the value of the signal parameter in a central database, using the value of the signal parameter that is maintained in the central database in computer code for a logic design forming part of an electrical circuit that includes the multiple bit signal, receiving an update to

the value of the signal parameter in the central database, updating the logic design with the updated value of the signal parameter by modifying the logic design to be compatible with the updated value of the signal parameter, and indicating design discrepancies occurring in the logic design that result from updating the value of the defined signal parameter automatically. The value of the signal parameter is maintained separately from the computer code for the logic design.

7.45. Panchul, Yamagishi, and IEEE Std. 1364-1995 neither describe nor suggest modifying a logic design to be compatible with an updated value of the signal parameter, where the value of the signal parameter is maintained separately from the computer code for the logic design, as recited in claim 18.

7.45.1. The Examiner respectfully replies:

7.45.2. The Examiner respectfully disagrees with the Applicant's assertion that Panchul, Yamagishi, and IEEE Std. 1364-1995 neither describe nor suggest modifying a logic design to be compatible with an updated value of the signal parameter, where the value of the signal parameter is maintained separately from the computer code for the logic design. As described in the rejection below, Panchul appears to teach modifying a logic design to be compatible with an updated value of the signal parameter. Further, under a broad interpretation, in Panchul figure 18B1, the labels ADDRESS_SIZE and DATA_SIZE are separate from the logic design since they are part of the "Global macro definitions", and not part of a module that contains logic design. Therefore, Panchul appears to teach the limitation that the value of the signal parameter is maintained separately from the computer code for the logic design. However, better references are used in the rejection below to teach the amended limitation.

7.46. The Applicant argues:

7.47. In this regard, as discussed above, Panchul's preliminary hardware designs are understood to be self-contained in standard high-level programming language.

Panchul's HDL designs and gate-level hardware representations are produced from the self-contained preliminary hardware designs and also believed to be self-contained.

7.48. Thus, Panchul neither describes nor suggests modifying a logic design to be compatible with an updated value of the signal parameter, where the value of the signal parameter is maintained separately from the computer code for the logic design, as recited in claim 18.

7.48.1. The Examiner respectfully replies:

7.48.2. The Examiner respectfully disagrees with the Applicant's conclusion that Panchul neither describes nor suggests modifying a logic design to be compatible with an updated value of the signal parameter, where the value of the signal parameter is maintained separately from the computer code for the logic design. The conclusion does not appear to follow from the premise. Further, as described in the rejection below, Panchul appears to teach modifying a logic design to be compatible with an updated value of the signal parameter. Further, under a broad interpretation, the labels ADDRESS_SIZE and DATA_SIZE are maintained separate from the logic design since they are part of the "Global macro definitions", and not part of a module that contains logic design. Therefore, Panchul appears to teach the limitation that the value of the signal parameter is maintained separately from the computer code for the logic design. However, better references are used in the rejection below to teach the amended limitation.

7.49. The Applicant argues:

7.50. Yamagishi and the IEEE Std. 1364-1995 do not remedy these deficiencies in Panchul. Indeed, Yamagishi at least suggests that an exhaustive array of different design data be maintained in a single "design entity," while the cited portion of IEEE Std. 1364-1995 deals with specifications of ranges of repetitive instances in the code that constitutes a hardware design. Accordingly, nothing in Yamagishi and the IEEE Std. 1364-1995 would lead one of ordinary skill to modify a logic design to be compatible with an

updated value of the signal parameter, where the value of the signal parameter is maintained separately from the computer code for the logic design, as recited in claim 18.

7.50.1. The Examiner respectfully replies:

7.50.2. The previous Office Action does not appear to rely upon IEEE and Yamagishi to teach the recited limitation.

7.51. The Applicant argues:

7.52. Thus, even if Panchul, Yamagishi, and the IEEE Std. 13641995 were combined, one of ordinary skill would not arrive at the recited subject matter. Accordingly, claim 18 is not obvious over. Panchul, Yamagishi, and the IEEE Std. 1364-1995. Applicant respectfully requests that the rejections of claim 18 and the claims dependent therefrom be withdrawn.

7.52.1. The Examiner respectfully replies:

7.52.2. A new rejection is made below to teach the amended limitation.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the

obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

10. Claims 1, 4 - 6, 7, 10 - 13, 15, 18, 21 - 23 and **25 - 27** are rejected under 35 U.S.C. 103(a) as being unpatentable over Panchul (U.S. Patent Number 6,226,776) in view of Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference), further in view of IEEE Verilog (IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language, 1995, IEEE).

10.1. The art of Panchul is directed to a computer aided hardware design system ***(Abstract)***.

10.2. The art of Yamagishi is directed to logic design systems ***(page 13.2.1, Abstract, and section 1 Introduction)***.

10.3. The art of IEEEVerilog is directed to logic design ***(page iii, section Introduction)***.

10.4. The art of Panchul and the art of Yamagishi are analogous art because they are both directed to a hardware design system.

10.5. The art of Panchul and the art of IEEEVerilog are analogous art because they both contain the art of logic design.

10.6. Regarding **claim 1**:

10.7. Panchul appears to teach:

10.7.1. a logic design module operable on one or more data processing devices to be used by one or more users to generate a logic design as part of an electrical circuit, wherein the logic design includes labels ***(figure 18B1, it would have been obvious that a logic design module was used to build the displayed code, wherein the logic design module was a tool such as SILOS III from Simucad, or a text editor)***.

10.7.2. a collection of modifiable values of signal parameters that are accessible by the logic design module, wherein the values of the signal parameters are associated with the labels in the logic design ~~and the collection of modifiable values is separate from the logic design~~ ***(figure 18B1, it would have been obvious that the parameters ADDRESS_SIZE and DATA_SIZE were signal parameters, and that they were accessible)***.

10.7.3. The logic design module is operable to update the logic design to reflect modification of the values of the signal parameters by modifying the logic design

to be compatible with the modified values of the signal parameters (*figure 18B1, it would have been obvious that compiling the Verilog code would have used the signal parameters to modify the signal widths, such as out_data, to be compatible with the modified values of the signal parameters*).

10.8. Panchul does not specifically teach:

10.8.1. *a central database integrated with the logic design module*.

10.8.2. The logic design module is operable to update the logic design to reflect modification of the signal parameters *in the central database*.

10.8.3. ~~a collection of modifiable values of signal parameters that are accessible by the logic design module, wherein the values of the signal parameters are associated with the labels in the logic design~~ and *the collection of modifiable values is separate from the logic design*.

10.8.4. to indicate design discrepancies in the logic design resulting from the modifications to the values of the signal parameters in the central database automatically.

10.9. Yamagishi appears to teach:

10.9.1. a central database integrated with the logic design module (*pages 13.2.2 and 13.2.3, section 2.2 Database FALNET*).

10.10. IEEEVerilog appears to teach:

10.10.1. a collection of modifiable values is separate from the logic design (*page 224, section 16.5 'include, entire first paragraph, "The 'include compiler directive can be used to include global or commonly used definitions and tasks without encapsulating repeated code within the module boundaries", and "Advantages of using the 'include compiler directive include the following: - Providing an integral part of configuration management – improving the organization of Verilog HDL source descriptions – facilitating the maintenance of Verilog HDL source descriptions; further, page 220 – 221 shows examples of definitions [see "commonly used definitions" above] using the 'define directive, and shows an example on page 221 first example, which shows a modifiable signal parameter used to define a signal width, as also taught by Panchul above; Further, please note that Panchul also teaches a collection of modifiable values separate from computer code in figure 15A at the top, "#include <stdarg.h>"; therefore, it would have been obvious to the ordinary artisan to put the collection of modifiable values in an 'include file*);

10.10.2. to indicate design discrepancies in the logic design resulting from the modifications to the values of the signal parameters in the central database automatically (*page 59, lines 1 – 25*).

10.10.2.1. Regarding (*page 59, lines 1 - 25*); it would have been obvious that updating a signal bit width incorrectly for terminal connections would result in too few or too many bits to connect all the instances, which would cause an error.

10.11. The motivation to use the art of Yamagishi with the art of Panchul would have been obvious given the statement in Yamagishi that the database allows much closer integration among tools than other commercially available frameworks *(page 13.2.3, left-side of page, lines 1 - 3)*, which would have been recognized as a benefit by the ordinary artisan.

10.12. The motivation to use the art of IEEEVerilog with the art of Panchul would have been the knowledge of the ordinary artisan that indicating design discrepancies would save time and effort by allowing the ordinary artisan to correct the discrepancies and thereby make the logic design functional. Further motivation would have been the benefits recited on page 224, section 16.5, including the recited advantages of improving the organization of Verilog HDL source descriptions and facilitating maintenance of Verilog HDL source descriptions.

10.13. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Yamagishi and the art of IEEEVerilog with the art of Panchul to produce the claimed invention.

==========

10.14. Regarding **claim 4**:

10.15. Panchul appears to teach indicating a bit width ***(figure 18B1, input signal in_unit has a bit width)***.

10.16. Panchul does not specifically teach indicating a bit width ***error***.

10.17. IEEEVerilog appears to teach indicating a bit width error ***(page 59, lines 1 – 25)***.

==========

10.18. Regarding **claim 5**:

10.19. Panchul appears to teach:

10.19.1. the signal parameters characterize a signal bit width (***figure 18B1, input signal in_data, DATA_SIZE***).

==========

10.20. Regarding **claim 6**:

10.21. Panchul appears to teach:

10.21.1. the signal parameters characterize a signal bit position (***figure 18B1, input signal in_data, element [DATA_SIZE - 1 : 0]***).

==========

10.22. Regarding **claim 26**:

10.23. Panchul appears to teach:

10.23.1. the signal parameters define characteristics that characterize multiple bits of a multiple bit signal (***figure 18B1, input signal in_data, element [DATA_SIZE - 1 : 0]***).

==========

10.24. Regarding **claim 27**:

10.25. Panchul appears to teach:

10.25.1. the signal parameter defines a characteristic that characterizes multiple bits of a multiple bit signal (***figure 18B1, input signal in_data, element [DATA_SIZE - 1 : 0]***).

==========

10.26. Regarding **claim 7**:

10.27. Panchul appears to teach:

10.27.1. receiving an assignment of a value to a signal parameter ***(figure 18B1; it would have been obvious that element DATA_SIZE received an assigned value)***

10.27.2. Maintaining the value of the signal parameter in association with an identifier of the signal parameter ***(figure 18B1; it would have been obvious that the value of a signal parameter was maintained – for example, DATA_SIZE)***.

10.27.3. Using the identifier of the signal parameter to identify a first position in computer code for a logic design forming part of an electrical circuit ***(figure 18B1, element DATA_SIZE; it would have been obvious that when the code was compiled that the Verilog compiler would use the identifier of the signal parameter to identify all positions in the computer code where the identifier was used)***.

10.27.4. Modifying the computer code at the first position to reflect the value (***figure 18B1, element DATA_SIZE; it would have been obvious that when the code was compiled that the Verilog compiler would modify the computer code at all positions to reflect the value***).

10.27.5. Using the identifier of the signal parameter to identify a second position in computer code for the logic design *(figure 18B1, element DATA_SIZE; it would have been obvious that when the code was compiled that the Verilog compiler would use the identifier of the signal parameter to identify all positions in the computer code where the identifier was used)*.

10.27.6. Receiving an updated value of the signal parameter *(figure 18B1, element DATA_SIZE; it would have been obvious to receive an updated value of the signal parameter through a logic design tool such as a text editor)*.

10.27.7. Modifying the computer code at the second position to reflect the value *(figure 18B1, element DATA_SIZE; it would have been obvious that when the code was compiled that the Verilog compiler would modify the computer code at all positions to reflect the value)*.

10.27.8. Updating both the first position and the second position in the computer code for the logic design to reflect the updated value of the signal parameter *(figure 18B1, element DATA_SIZE; it would have been obvious that when the code was compiled that the Verilog compiler would update all positions in the computer code for the logic design to reflect the updated value of the signal parameter)*.

10.28. Panchul does not specifically teach:

10.28.1. Maintaining the defined signal value ***in a central database***.

10.28.2. Using the identifier of the signal parameter maintained ***in the central database*** .

10.28.3. Receiving an updated value of the signal parameter ***in the central database***.

10.28.4. indicating design discrepancies occurring in the logic design that result from updating the value of the defined signal parameter automatically.

10.28.5. wherein the value of the signal parameter is maintained separately from the computer code for the logic design.

10.29. Yamagishi appears to teach:

10.29.1. a central database integrated with a logic design module ***(pages 13.2.2 and 13.2.3, section 2.2 Database FALNET)***.

10.30. IEEEVerilog appears to teach:

10.30.1. indicating design discrepancies occurring in the logic design that result from updating the value of the defined signal parameter automatically (*page 59, lines 1 – 25*).

10.30.1.1. Regarding (*page 59, lines 1 – 25*); it would have been obvious that updating a signal bit width incorrectly for terminal connections would result in too few or too many bits to connect all the instances, which would cause an error.

10.30.2. wherein the value of the signal parameter is maintained separately from the computer code for the logic design (*page 224, section 16.5 'include, entire first paragraph, "The 'include compiler directive can be used to include global or commonly used definitions and tasks without encapsulating repeated code within the module boundaries", and "Advantages of using the 'include compiler directive include the following: - Providing an integral part of configuration management – improving the organization of Verilog HDL source descriptions – facilitating the maintenance of Verilog HDL source descriptions; further, page 220 – 221 shows examples of definitions [see "commonly used definitions" above] using the 'define directive, and shows an example on page 221 first example, which shows a modifiable signal parameter used to define a signal width, as also taught by Panchul above; therefore, it would have been obvious to the ordinary artisan to put the signal parameters in an 'include file*);

10.31. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Yamagishi and the art of IEEEVerilog with the art of Panchul to produce the claimed invention.

10.32. Regarding **claim 11**:

10.32.1. Panchul appears to teach:

10.32.1.1. the signal parameter characterizes a signal bit width and the value includes a value for the signal bit width (***figure 18B1, input signal in_data, element [DATA_SIZE – 1 : 0]***).

10.33. Regarding **claim 12**:

10.33.1. Panchul appears to teach:

10.33.2. the signal parameter characterizes a signal bit position and the value includes a value for the signal bit position (***figure 18B1, input signal in_data, element [DATA_SIZE – 1 : 0]***).

10.34. Regarding **claim 13**:

10.34.1. Panchul appears to teach:

10.34.2. the signal parameter characterizes a bit field and the value includes a value for the bit field (*figure 18B1, input signal in_data, element [DATA_SIZE – 1 : 0]*).

==========

10.35. Regarding **claim 10**:

10.35.1. Panchul does not specifically teach:

10.35.1.1. graphically indicating a bit width error.

10.35.2. Yamagishi appears to teach graphically indicating bit widths (**figure 1**).

10.35.3. Yamagishi appears to teach graphically indicating errors (**figure 1**).

==========

10.36. Regarding **claim 18**:

10.37. Panchul appears to teach:

10.37.1. Receiving a value of a signal parameter that characterizes multiple bits of a multiple bit signal (*figure 18B1, element DATA_SIZE; it would have*

been obvious that a value was received in order to have a value of the parameter).

10.37.2. Maintaining the value of the signal parameter *(figure 18B1, element DATA_SIZE)*.

10.37.3. Using the value of the signal parameter that is maintained, in computer code for a logic design forming part of an electrical circuit that includes the multiple bit signal *(figure 18B1, element DATA_SIZE)*.

10.37.4. Receiving an update to the value of the signal parameter (*figure 18B1, element DATA_SIZE; it would have been obvious to receive an updated value of the signal parameter through a logic design tool such as a text editor*).

10.37.5. updating the logic design with the updated value of the signal parameter by modifying the logic design to be compatible with the updated value of the signal parameter (*figure 18B1, element DATA_SIZE; it would have been obvious that compiling the Verilog code would have updated the logic design by modifying the logic design to be compatible with the updated value of the signal parameter*).

10.38. Panchul does not teach specifically teach:

10.38.1. Using the defined signal parameter ***that is maintained in the central database*** in computer code for a logic design forming part of an electrical circuit.

10.38.2. Maintaining the value of signal parameter ***in a central database***.

10.38.3. Using the value of the signal parameter that is maintained ***in the central database***.

10.38.4. indicating design discrepancies occurring in the logic design that result from updating the value of the defined signal parameter automatically.

10.38.5. wherein the value of the signal parameter is maintained separately from the computer code for the logic design.

10.39. Yamagishi appears to teach:

10.39.1. Maintaining logic design data ***in a central database (page 13.2.1, section 2 FALcyber, and pages 13.2.2 and 13.2.3, section 2.2 Database FALNET)***.

10.40. IEEEVerilog appears to teach:

10.40.1. indicating design discrepancies occurring in the logic design that result from updating the value of the defined signal parameter automatically (*page 59, lines 1 - 25*).

10.40.2. wherein the value of the signal parameter is maintained separately from the computer code for the logic design (*page 224, section 16.5 'include, entire first paragraph, "The 'include compiler directive can be used to include global or commonly used definitions and tasks without encapsulating repeated code within the module boundaries", and "Advantages of using the 'include compiler directive include the following: - Providing an integral part of configuration management – improving the organization of Verilog HDL source descriptions – facilitating the maintenance of Verilog HDL source descriptions; further, page 220 – 221 shows examples of definitions [see "commonly used definitions" above] using the 'define directive, and shows an example on page 221 first example, which shows a modifiable signal parameter used to define a signal width, as also taught by Panchul above; therefore, it would have been obvious to the ordinary artisan to put the signal parameter values in an 'include file*);

10.41. The motivation to use the art of Yamagishi with the art of Panchul would have been obvious given the statement in Yamagishi that the database allows much closer integration among tools than other commercially available frameworks (*page 13.2.3, left-side of page, lines 1 – 3*), which would have been recognized as a benefit by the ordinary artisan.

10.42. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Yamagishi and the art of IEEEVerilog with the art of Panchul to produce the claimed invention.

10.43. Regarding **claim 22**:

10.43.1. Panchul appears to teach:

10.43.1.1. the signal parameter characterizes a signal bit width and the value includes a value for the signal bit width (***figure 18B1, input signal in_data, element [DATA_SIZE - 1 : 0]***).

10.44. Regarding **claim 23**:

10.44.1. Panchul appears to teach:

10.44.1.1. the signal parameter characterizes a signal bit position and the value includes a value for the signal bit position (***figure 18B1, input signal in_data, element [DATA_SIZE - 1 : 0]***).

10.45. Regarding **claim 25**:

10.45.1. Panchul does not specifically teach:

10.45.1.1. permitting one or more users to access the central database.

10.45.2. Yamagishi appears to teach:

10.45.2.1. permitting one or more users to access the central database *(page 13.2.1, figure 1)*.

10.46. Regarding **claim 21**:

10.46.1. Panchul does not specifically teach:

10.46.1.1. graphically indicating a bit width error.

10.46.2. Yamagishi appears to teach graphically indicating bit widths **(figure 1)**.

10.46.3. Yamagishi appears to teach graphically indicating errors **(figure 1)**.

==========

10.47. Regarding **claim 15**:

10.48. Panchul appears to teach:

10.48.1. a collection of identifiers of one or more bit width signal parameters (*figure 18B1, elements DATA_SIZE and ADDRESS_SIZE*).

10.48.2. values associated with each of the identifiers of the bit width signal parameters *figure 18B1, elements DATA_SIZE 32 and ADDRESS_SIZE 16)*.

10.48.3. modification logic operable on one or more data processing devices to allow a user to modify the values associated with the identifiers individually (*figure 18B1; it would have been obvious that a tool such as a text editor would allow the user to modify the values associated with the identifiers*).

10.48.4. an interface to convey the identifiers and the associated values to a logic design module that is operable on one or more data processing devices and that uses the identifiers to identify where a logic design is to be changed and the values to change a bit width in the logic design to form part of an electrical circuit (*figure 18B1, elements DATA_SIZE and ADDRESS_SIZE; it would have been obvious that there was an interface, such as a Verilog compiler, that used the identifiers to identify where the logic design was to be changed*), wherein the collection of identifiers of one or more bit width signal parameters is maintained (*figure 18B1; it would have been obvious that a tool such as a text editor would*

allow the user to modify the values associated with the identifiers) ~~separately from the logic design~~;

10.49. Panchul does not specifically teach:

10.49.1. *A central database accessible by one or more users.*

10.49.2. One or more signal parameters defined *in the central database*.

10.49.3. an interface to convey the identifiers and the associated values from *the central database*.

10.49.4. ~~wherein the collection of identifiers of one or more bit width signal parameters is maintained~~ separately from the logic design.

10.50. Yamagishi appears to teach a central database accessible by one or more users *(page 13.2.1, figure 1, and pages 13.2.2 and 13.2.3, section 2.2 Database FALNET)*.

10.51. IEEEVerilog appears to teach:

10.51.1. ~~wherein the~~ *a* collection of identifiers of one or more ~~bit width signal~~ parameters is maintained separately from the logic design (*page 224, section 16.5 'include, entire first paragraph, "The 'include compiler directive can be used to include global or commonly used definitions and tasks without encapsulating repeated code within the module boundaries", and "Advantages of*

using the 'include compiler directive include the following: - Providing an integral part of configuration management – improving the organization of Verilog HDL source descriptions – facilitating the maintenance of Verilog HDL source descriptions; further, page 220 – 221 shows examples of definitions [see "commonly used definitions" above] using the 'define directive, and shows an example on page 221 first example, which shows a modifiable bit width signal parameter used to define a signal bit width, as also taught by Panchul above; therefore, it would have been obvious to the ordinary artisan to put the collection of identifiers of parameters in an 'include file);

10.52. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Yamagishi and the art of IEEEVerilog with the art of Panchul to produce the claimed invention.

11. Examiner's Note: Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner.

Conclusion

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

13. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

14. The prior art made of record and not relied upon is used to teach knowledge of the ordinary artisan at the time of invention:

14.1. Crosetto (U.S. Patent Number 7,051,309) teaches a configuration parameter file separate from a logic design containing identifiers and associated values of signal parameters used to define signal widths (*columns 25 – 28, table 1, configuration parameters; and table 3, teaches signals defined using the signal widths of table 1*).

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Russ Guill whose telephone number is 571-272-7955. The examiner can normally be reached on Monday – Friday 9:30 AM – 6:30 PM.
16. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature or relating to the status of this application should be directed to the TC2100 Group Receptionist: 571-272-2100.
17. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Russ Guill
Examiner
Art Unit 2123

RG

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100